

Міністерство освіти і науки України  
Харківський національний університет імені В. Н. Каразіна  
Навчально-науковий інститут комп'ютерних наук та штучного інтелекту  
Кафедра комп'ютерних систем та робототехніки

«Затверджую»

в.о. завідуючого кафедри

комп'ютерних систем та робототехніки

к. ф.-м. н., доцент Максим ХРУСЛОВ

«\_\_\_» червня 2025 р.

## Пояснювальна записка

до кваліфікаційної роботи  
бакалавра

на тему: «МОДЕЛЬ КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ  
ЕКОЛОГІЧНОГО СТАНУ РЕГІОНУ»

Спеціальність 123 – Комп'ютерна інженерія

Галузь знань: 12 – Інформаційні технології.

Освітня програма «Комп'ютерна інженерія».

Захищено на засіданні

Екзаменаційної комісії № 44

протокол № \_\_ від \_\_.06.2025 р.

Оцінка \_\_\_\_ / \_\_\_\_\_

Голова Екзаменаційної комісії

\_\_\_\_\_ ЧУГАЙ А. М.

Виконала:

Студентка групи КІ-41

АЗАРЕНКОВА Олександра Андріївна

Керівник:

к.т.н., доцент кафедри комп'ютерних  
систем та робототехніки

СТРІЛЕЦЬ Вікторія Євгенівна

Рецензент: к.т.н., доцент, в.о. завідувача  
кафедри теоретичної та прикладної  
інформатики

МЕНЯЙЛОВ Євген Сергійович

\_\_\_\_\_

Харків – 2025

## АНОТАЦІЯ

Пояснювальна записка до кваліфікаційної роботи бакалавра складається зі вступу, трьох розділів, висновків, списку використаних джерел і 4 додатків. Загальний обсяг роботи складає 57 сторінок, із яких 35 сторінок основної частини з 9 рисунками, 6 таблицями, списку використаних джерел із 20 найменувань та 4 додатками.

У роботі проведений аналіз задач екологічного моніторингу і розроблений прототип комп'ютерної системи моніторингу екологічного стану регіону або місцевості.

**Мета кваліфікаційної роботи** – підвищити якість прогнозування показників екологічного стану окремого регіону через застосування підходів штучного інтелекту та машинного навчання.

**Об'єкт дослідження** – процес моніторингу екологічного стану окремого регіону.

**Предмет дослідження** – моделі і методи аналізу показників екологічного стану, зокрема методи машинного навчання і нейронні мережі.

Модель комп'ютерної системи екологічного моніторингу може стати основою для створення інформаційних систем моніторингу довкілля, і бути корисною як навчальна та демонстраційна система для ознайомлення студентів із інтеграцією методів аналізу даних, машинного навчання, штучного інтелекту в галузь екології, охорони природи і довкілля.

**Ключові слова:** екологічний моніторинг, прогнозування, машинне навчання, комп'ютерні інформаційні системи.

## ABSTRACT

The explanatory note to the bachelor's qualification work consists of an introduction, three chapters, conclusions, a list of references, and 4 appendices. The total volume of the work is 57 pages, including 35 pages of the main part with 9 figures, 6 tables, a list of 20 references, and 4 appendices.

The research analyses the tasks of environmental monitoring and develops a prototype of a computer system for monitoring the environmental condition of a region or area.

**The aim of the qualification work** is to improve the accuracy of forecasting environmental indicators in a specific region by applying artificial intelligence and machine learning approaches.

**The object of the study** is the process of environmental monitoring in a particular region.

**The subject of the study** is models and methods for analyzing environmental indicators, particularly machine learning methods and neural networks.

The model of a computer-based environmental monitoring system can become the basis for the creation of environmental monitoring information systems, and can be useful as a training and demonstration system to introduce students to the integration of data analysis, machine learning, and artificial intelligence methods in the field of ecology, nature and environmental protection.

**Keywords:** *environmental monitoring, forecasting, machine learning, computer information systems.*

## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ.....	6
ВСТУП .....	7
РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ТА СИСТЕМ ЕКОЛОГІЧНОГО МОНІТОРИНГУ .....	9
1.1 Задачі екологічного моніторингу .....	9
1.2 Проблеми в галузі екологічного моніторингу .....	11
1.3 Інформаційні системи моніторингу довкілля .....	13
1.4 Методи екологічного моніторингу.....	18
Висновки до розділу 1 .....	20
РОЗДІЛ 2 ПРОЄКТУВАННЯ МОДЕЛІ КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ ЕКОЛОГІЧНОГО СТАНУ .....	21
2.1 Вимоги до комп'ютерної системи екологічного моніторингу.....	21
2.2 Модель архітектури комп'ютерної системи екологічного моніторингу ....	23
2.3 Модель структури бази даних комп'ютерної системи.....	25
2.4 Методи аналізу даних моніторингу екологічного стану .....	26
Висновки за розділом 2 .....	28
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТА ЇЇ ТЕСТУВАННЯ.....	29
3.1 Обґрунтування вибору програмних технологій .....	29
3.2 Збір і збереження даних моніторингу .....	31
3.2.1 Джерела і збір даних моніторингу .....	31
3.2.2 Програмна реалізація бази даних .....	32
3.3 Аналіз даних моніторингу екологічного стану регіону.....	33
3.4 Тестування комп'ютерної системи моніторингу екологічного стану .....	37
Висновки за розділом 3 .....	37
ВИСНОВКИ.....	40

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	41
ДОДАТКИ.....	44
Додаток А.....	44
Додаток Б.....	46
Додаток В.....	49
Додаток Г.....	54

## ПЕРЕЛІК СКОРОЧЕНЬ І УМОВНИХ ПОЗНАЧЕНЬ

ГДК – гранично-допустима концентрація;

API – Application Programming Interface, прикладний програмний інтерфейс;

KPI – Key Performance Indicators, ключові показники ефективності;

AQI – Air quality index, індекс якості повітря;

ML – Machine Learning, машинне навчання.

## ВСТУП

У сучасних умовах зростання техногенного навантаження на довкілля питання екологічної безпеки набуває особливої актуальності. Для прийняття ефективних рішень щодо розробки і впровадження програм, засобів або технологій, пов'язаних із зменшенням забруднення навколишнього середовища, необхідно мати достовірну, актуальну та систематизовану інформацію про екологічний стан певного спостережуваного регіону. Розробка комп'ютерних систем моніторингу екологічного стану регіону є важливим напрямом у сфері охорони навколишнього середовища, і саме розвиток інформаційних технологій, комп'ютерної інженерії дає можливість удосконалення та оптимізації багатьох процесів.

Інформаційні технології дозволяють автоматизувати процеси збору, обробки, зберігання та візуалізації екологічних даних. Впровадження систем моніторингу з використанням сенсорних мереж, Інтернету речей (IoT) та сучасних методів обробки даних (у тому числі з використанням хмарних сервісів та машинного навчання) забезпечує нові можливості для аналізу екологічної ситуації.

**Метою кваліфікаційної роботи** є підвищення якості прогнозування показників екологічного стану окремого регіону через застосування підходів штучного інтелекту та машинного навчання.

**Об'єкт дослідження** – процес моніторингу екологічного стану окремого регіону.

**Предмет дослідження** – моделі і методи аналізу показників екологічного стану, зокрема методи машинного навчання і нейронні мережі.

**Методи дослідження** – методи системного аналізу, проєктування програмного забезпечення, моделювання, статистичного аналізу, машинного навчання.

Для досягнення мети необхідно виконати такі **завдання**:

1. Провести аналіз існуючих рішень у сфері екологічного моніторингу.
2. Сформулювати вимоги до комп'ютерної системи екологічного моніторингу.
3. Розробити модель комп'ютерної системи моніторингу екологічного стану.
4. Програмно реалізувати прототип системи.
5. Провести тестування та оцінити результати її роботи комп'ютерної системи.

# РОЗДІЛ 1

## АНАЛІЗ ІСНУЮЧИХ ПІДХОДІВ ТА СИСТЕМ ЕКОЛОГІЧНОГО МОНІТОРИНГУ

### 1.1 Задачі екологічного моніторингу

*Екологічний моніторинг* – це система спостережень, оцінки та прогнозу змін стану навколишнього середовища під впливом як природних, так і антропогенних факторів. Основна мета моніторингу — своєчасне виявлення негативних змін і запобігання екологічним загрозам.

До основних задач екологічного моніторингу належать:

1) збір актуальної інформації про стан довкілля: вимірювання параметрів повітря, води, ґрунтів (температура, вологість, концентрація шкідливих речовин); фіксація часових та просторових змін;

2) оцінка рівня забруднення навколишнього середовища: визначення концентрацій шкідливих речовин (CO<sub>2</sub>, PM<sub>2.5</sub>, SO<sub>2</sub>, NO<sub>x</sub> тощо); порівняння фактичних значень з нормативними межами (ГДК);

3) виявлення джерел забруднення: географічна локалізація промислових підприємств, транспорту, звалищ; аналіз впливу джерел на навколишнє середовище;

4) прогнозування розвитку екологічної ситуації: побудова моделей зміни екологічних параметрів у часі; Виявлення тенденцій погіршення або поліпшення стану;

5) інформування органів влади та населення: надання звітів, графіків, аналітики для прийняття рішень; оповіщення у разі перевищення допустимих норм забруднення.

6) підтримка прийняття управлінських рішень у галузі екології: формування рекомендацій щодо екологічної політики; оцінка ефективності заходів із захисту довкілля.

7) аналіз історичних даних: збереження бази даних спостережень для довготривалого аналізу; вивчення впливу сезонності, кліматичних умов, людської діяльності.

Задачі екологічного моніторингу значною мірою залежать від *природно-географічних, економічних і соціальних особливостей регіону*. Наприклад, в Україні можна виділити такі екологічні регіони:

- індустріальні регіони (Дніпропетровська, Донецька обл.): основна увага зосереджена на викидах промисловості, забрудненні повітря, ґрунтів, вод;
- аграрні регіони (Вінницька, Полтавська обл.): моніторинг впливу добрив, пестицидів, забруднення водоносних горизонтів;
- карпатський регіон: фокус на збереженні лісів, біорізноманіття, стану водних ресурсів;
- приморські регіони (Одеська, Миколаївська, Херсонська обл.): основні напрямки – контроль якості вод Чорного моря, стану лиманів, берегової ерозії;
- міські агломерації (Київ, Харків, Львів): пріоритетом є якість атмосферного повітря, шумове забруднення, транспортний вплив.

Екологічний моніторинг в Україні здійснюється відповідно до низки законів, постанов та міжнародних зобов'язань. Серед *основних нормативних документів* варто зазначити:

Закон України «Про охорону навколишнього природного середовища» визначає основи організації моніторингу, повноваження органів влади та права громадян.

Закон України «Про екологічний моніторинг» (проект, наразі регулюється підзаконними актами) призначений для систематизації підходів до державного моніторингу.

Постанова КМУ № 1376 (2001) «Про затвердження Положення про здійснення державного моніторингу навколишнього природного середовища»

описує структуру моніторингу, розподіл зон спостереження та відповідальних суб'єктів.

Національні стандарти ДСТУ та ГДК (гранично допустимі концентрації) регламентують порогові значення для шкідливих речовин у повітрі, воді, ґрунті.

Постанова КМУ № 827 (2017) «Про Єдину державну систему моніторингу довкілля» містить вимоги до інтеграції даних від різних установ, створення відкритих платформ.

Також Україна має ратифікувала міжнародні конвенції та має певні міжнародні зобов'язання:

- Організація Об'єднаних Націй (Цілі сталого розвитку) – цілі 6, 11, 13 та 15.
- Орхуська конвенція – право громадян на доступ до екологічної інформації.
- інтеграція з європейською системою Copernicus (для супутникового моніторингу та аналітики).

## **1.2 Проблеми в галузі екологічного моніторингу**

Основними проблемами екологічного моніторингу в Україні та світі залишаються:

### **1. Нерівномірне охоплення територій системами моніторингу.**

Більшість пунктів (місць) спостереження зосереджені у великих містах та промислових регіонах, тоді як сільські, гірські, прибережні та прикордонні зони часто залишаються поза зоною регулярного спостереження. У таких умовах:

- неможливо швидко виявити локальні забруднення або витoki;
- відсутність даних не дозволяє побудувати повноцінну картину екологічного стану;

– вразливі екосистеми (наприклад, заповідні території) залишаються без захисту.

## 2. Застаріле обладнання та методи вимірювань.

Багато стаціонарних пунктів моніторингу використовують пристрої, вироблені ще у 1980–1990-х роках, або обладнання без регулярного технічного обслуговування. Це призводить до:

- низької точності вимірювань;
- відсутності сучасних сенсорів для виявлення нових типів забруднювачів (наприклад, мікропластику або летких органічних сполук);
- неможливості інтеграції з цифровими платформами без ручного втручання.

Також в багатьох випадках застосовуються застарілі методики аналізу, які не відповідають сучасним міжнародним стандартам (наприклад, ISO, EPA тощо).

## 3. Ручна або неавтоматизована обробка даних.

Обробка інформації часто виконується вручну або з використанням неспеціалізованих програм, які не призначені для великих масивів екологічних даних. Це має такі наслідки:

- суттєві затримки між збором даних і їх публікацією;
- вища ймовірність людських помилок;
- неможливість реалізувати оперативне реагування на перевищення допустимих значень;
- недостатня швидкість прийняття рішень на державному чи місцевому рівнях.

## 4. Відсутність єдиного інформаційного простору.

Екологічні дані збираються різними установами – Держекоінспекцією, гідрометеоцентрами, місцевими лабораторіями, науковими інститутами – і,

відповідно, окремо обробляються. Відсутність централізованої, синхронізованої бази призводить до:

- дублювання даних або, навпаки, їх втрати;
- неможливості порівняння результатів із різних джерел;
- низької доступності інформації для громадськості, науковців і бізнесу;
- обмежень у впровадженні сучасних інструментів візуалізації (наприклад, інтерактивних карт чи мобільних застосунків).

Вирішення цих проблем можливе шляхом впровадження автоматизованих систем екологічного моніторингу, що використовують комп'ютерні технології для збору, зберігання, обробки та представлення даних.

### **1.3 Інформаційні системи моніторингу довкілля**

На сьогодні у світі існує багато рішень для моніторингу навколишнього середовища. Розглянемо кілька найвідоміших.

**AirVisual (IQAir)** – це глобальна платформа для відстеження стану повітря в реальному часі. Розроблена компанією IQAir (Швейцарія) і відома у сфері рішень для очищення повітря та моніторингу його якості.

AirVisual агрегує дані з офіційних джерел (метеостанцій, урядових агентств) та з тисяч особистих пристроїв користувачів (наприклад, AirVisual Node або AirVisual Pro). Дані передаються через інтернет на хмарний сервер, обробляються, калібруються, зводяться в єдину модель і виводяться на інтерактивну карту на сайті (рис. 1.1), у мобільний застосунок або API для розробників.

Основні функції:

- відображення рівня PM2.5, PM10, CO<sub>2</sub>, температури та вологості;
- прогноз якості повітря на 7 днів;
- порівняння даних між регіонами;
- можливість публікувати свої дані.

Обмеження:

- висока вартість власних сенсорів IQAir;
- приватні дані не завжди стандартизовані — можливі похибки;
- потребує постійного підключення до Інтернету;
- залежність від хмарної інфраструктури.

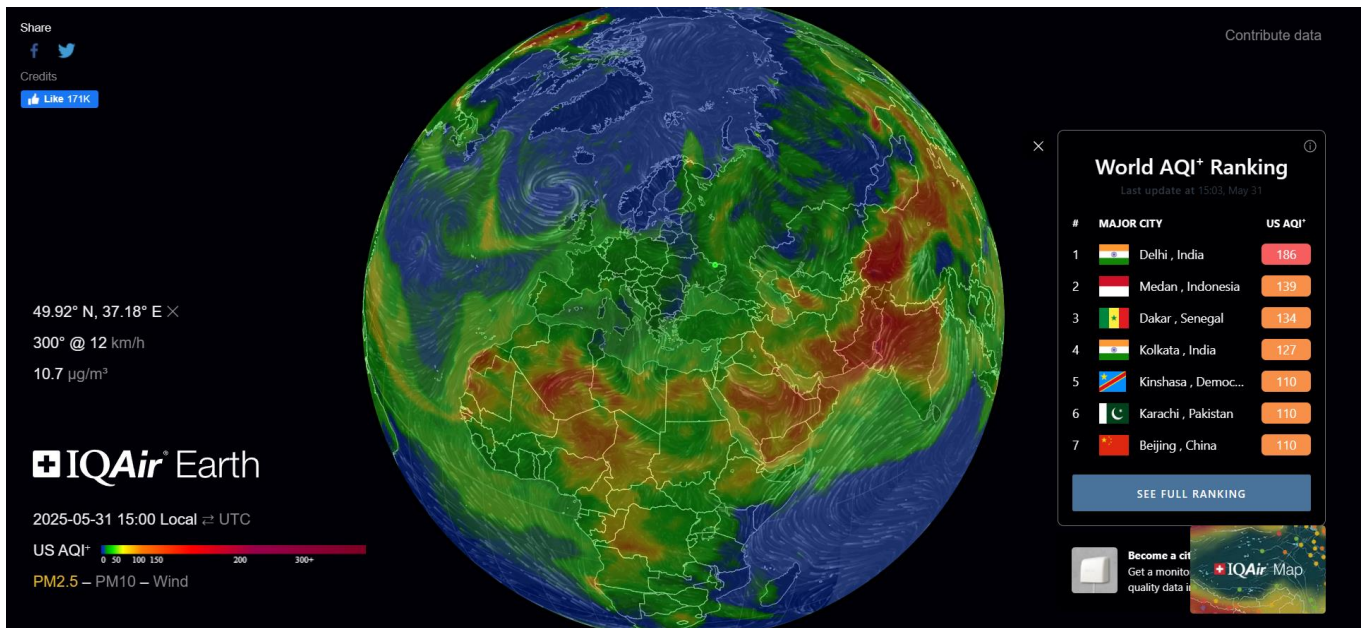


Рисунок 1.1 – Мапа забруднення повітря на Землі IQAir

**PurpleAir** – система від американської компанії PurpleAir, яка зосереджується на демократизації екологічного моніторингу через дешеві сенсори.

Кожен пристрій PurpleAir оснащений лазерними сенсорами для вимірювання концентрації твердих частинок (PM1.0, PM2.5, PM10). Дані передаються через Wi-Fi на відкритий сервер компанії та стають доступними на мапі в реальному часі (рис. 1.2).

Система не потребує складного налаштування – під'єднання до Wi-Fi та реєстрація на сайті.

Основні функції:

- моніторинг твердих частинок у повітрі;
- публікація даних на інтерактивній карті;
- відкритий API для дослідників;
- підтримка калібрування для точніших результатів.

Обмеження:

- обмежений набір параметрів (немає CO<sub>2</sub>, SO<sub>2</sub>, температури);
- залежність від Wi-Fi та електроживлення;
- дані з окремих сенсорів можуть мати похибку без калібрування;
- встановлення лише у доступних локаціях — уникає «мертвих зон».

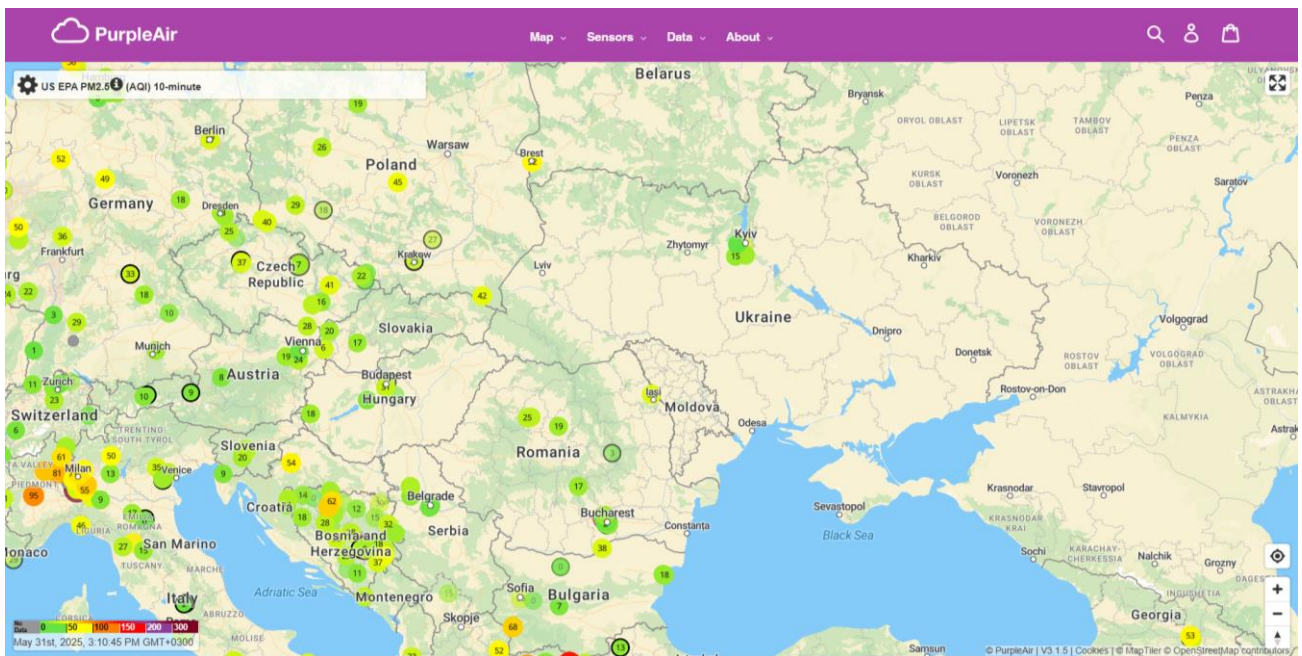


Рисунок 1.2 – Мапа концентрації твердих частинок у повітрі від PurpleAir

**Automatic Weather Stations (AWS)** – великі метеостанції. Найвідоміші виробники метеорологічного обладнання: Davis Instruments, Vaisala, Campbell Scientific, а також локальні інтегратори у аграрному секторі.

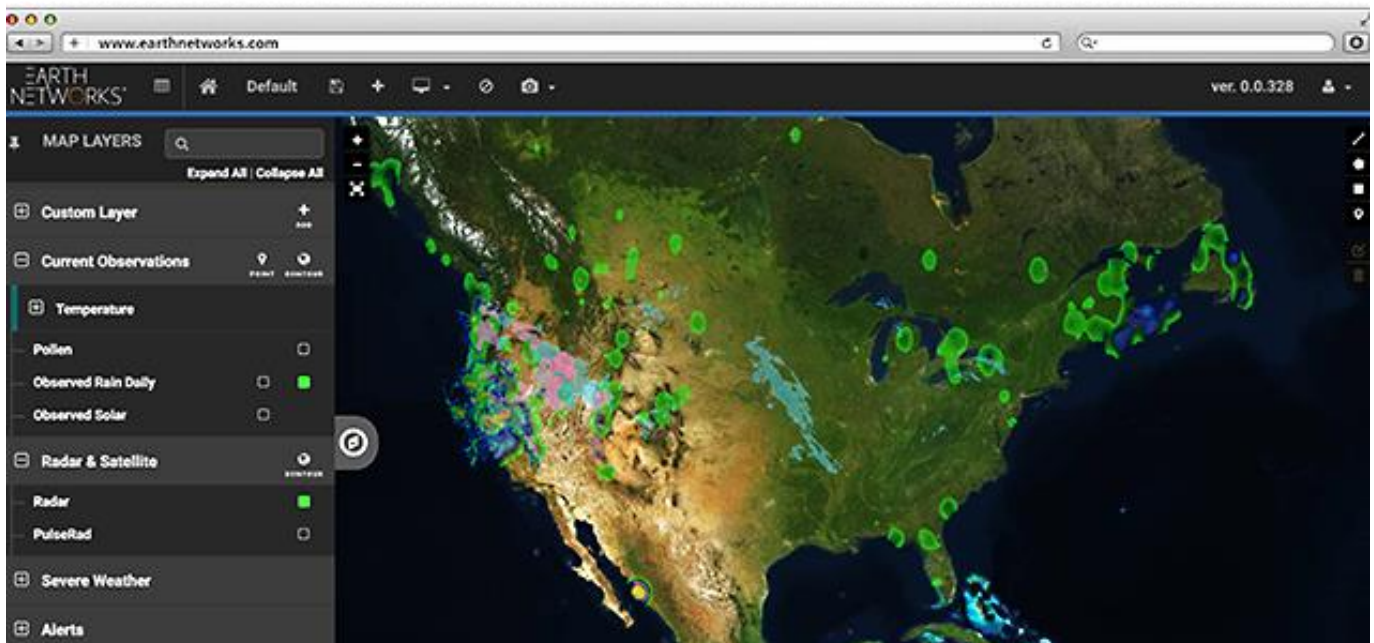
Метеостанції встановлюються у полях або на промислових об'єктах. Вони збирають дані про погодні умови: температуру, вологість, швидкість та напрямок вітру, опади, інсоляцію. Можуть також мати додаткові модулі для ґрунтової вологи, ультрафіолетового випромінювання чи якості повітря. Передача даних здійснюється через GSM-модеми, LoRaWAN або Wi-Fi до серверів агроплатформ (наприклад, Meteotrek, AgroMonitor).

Основні функції:

- постійний моніторинг метеопоказників (рис. 1.3);
- прогнозування погодних ризиків для аграріїв;
- індивідуальні погодні моделі для полів;
- інтеграція з дронами, GPS, агросистемами.

Обмеження:

- висока вартість станцій;
- складність у встановленні й обслуговуванні;
- потрібне регулярне калібрування датчиків;
- метеостанції покривають лише невелику площу — потрібна мережа.



### Рисунок 1.3 – Мапи моніторингу від компанії EarthNetworks

**Регіональні системи екологічного моніторингу** розроблюються окремими компаніями під особливості регіону. В Україні – це здебільшого локальні ініціативи, наприклад ЛУН Місто Air, SaveEcoBot, ЕкоСіті, або муніципальні системи від обласних центрів.

Сенсори (такі, як Nova SDS011, MQ135) встановлюються активістами, громадськими організаціями чи місцевою владою. Дані збираються на локальний сервер або через хмарні сервіси (Firebase, AWS, Heroku). Часто застосовуються Arduino або ESP32 контролери з передачами по Wi-Fi / GSM.

Основні функції:

- моніторинг PM2.5, PM10, температури, вологості;
- відображення на локальній карті або Telegram-боті;
- можливість швидкого виведення даних у відкритий доступ;
- сповіщення про перевищення норм.

Обмеження:

- часто обмежений масштаб (одне місто чи регіон);
- відсутність централізованої перевірки даних;
- нетривка робота при поганому інтернет-з'єднанні;
- обмежений життєвий цикл сенсорів при роботі в агресивному середовищі.

Проектування і впровадження сучасних систем моніторингу здійснюється завдяки таким технологіям:

- сенсорні мережі (WSN) – автономні пристрої з датчиками для збору даних;
- інтернет речей (IoT) – забезпечує зв'язок між сенсорами та хмарними сервісами;

- хмарні платформи – AWS, Azure, Google Cloud для зберігання й обробки даних;
- геоінформаційні системи (ГІС) – для відображення даних на мапах;
- штучний інтелект і машинне навчання – аналіз даних і прогнозування змін екологічного стану;
- протоколи передачі даних – MQTT, HTTP, LoRaWAN тощо.

#### **1.4 Методи екологічного моніторингу**

Існує багато методів екологічного моніторингу, які використовуються для збору, обробки та аналізу інформації про стан довкілля. Вони поділяються на *інструментальні, біологічні та аналітичні* методи. Основну увагу в сучасних системах приділяють аналітичним методам, адже саме вони дозволяють не лише фіксувати стан навколишнього середовища, а й прогнозувати його зміни.

1. Інструментальні методи полягають у використанні датчиків, сенсорів і приладів (газоаналізатори, метеостанції, дрони) для моніторингу параметрів: температура, вологість, концентрація CO<sub>2</sub>, PM<sub>2.5</sub>, PM<sub>10</sub>, NO<sub>2</sub>, SO<sub>2</sub> тощо.

Переваги: висока точність, автоматизація, можливість безперервного збору даних.

Недоліки: висока вартість обладнання, потреба в технічному обслуговуванні.

2. Біологічні методи – це оцінка стану довкілля за змінами в популяціях живих організмів (біоіндикатори: лишайники, мохи, молюски). Використовуються для довготривалого оцінювання екосистем.

Недоліки: повільність, суб'єктивність оцінки, потреба в експертному супроводі.

3. Аналітичні методи – це обробка зібраних даних за допомогою математичних, статистичних та інтелектуальних алгоритмів (табл. 1.1). Вони

дають можливість виявляти закономірності, аномалії, тенденції, та прогнозувати майбутній стан довкілля.

*Таблиця 1.1*

**Аналітичні методи в екологічному моніторингу**

<b>Метод</b>	<b>Суть</b>	<b>Приклади використання</b>
Статистичний аналіз	Визначення середніх значень, відхилень, трендів	Тренди забруднення повітря за 10 років
Геостатистика (GIS)	Просторовий аналіз та картографування	Карти забруднення, аналіз «гарячих точок»
Машинне навчання	Побудова моделей прогнозу на основі історичних даних	Прогноз рівня забруднення за погодними умовами
Аналіз часових рядів	Вивчення змін показників у часі	Виявлення сезонних коливань забруднення
Спектральний аналіз	Аналіз складових частот в даних	Дослідження коливань параметрів на різних масштабах часу
Мультифакторний аналіз	Вивчення взаємозв'язку багатьох факторів	Вплив промисловості, транспорту, метеоумов одночасно

Перспективи використання та вдосконалення аналітичних методів:

1) інтеграція з штучним інтелектом (ШІ):

- використання нейронних мереж для прогнозування екологічних катастроф (пожеж, забруднень);

- інтелектуальний аналіз супутникових знімків (remote sensing + AI);

2) Edge-комп'ютинг та IoT: аналіз даних безпосередньо на пристроях збору (сенсори, дрони) без потреби передачі великих обсягів інформації;

3) глибокий аналіз Big Data: використання технологій Hadoop, Spark для роботи з великими обсягами екологічних даних;

4) моделювання сценаріїв: аналітичне моделювання розвитку подій (наприклад, як зміниться стан повітря при збільшенні трафіку або викидів).

### **Висновки до розділу 1**

Проведений аналіз показав, що сучасні комп'ютерні системи моніторингу екологічного стану дозволяють значно підвищити точність, швидкість і ефективність аналізу даних. Водночас більшість існуючих рішень або мають вузьке застосування, або не забезпечують повної автоматизації для локального рівня. Отже, доцільним є проєктування нової моделі комп'ютерної системи моніторингу, яка буде адаптована під специфіку обраного регіону, із можливістю масштабування, інтеграції з IoT-пристроями та надання даних у режимі реального часу.

## РОЗДІЛ 2

### ПРОЄКТУВАННЯ МОДЕЛІ КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ ЕКОЛОГІЧНОГО СТАНУ

#### 2.1 Вимоги до комп'ютерної системи екологічного моніторингу

Комп'ютерна система екологічного моніторингу повинна забезпечувати комплексний підхід до збору, обробки, зберігання та візуалізації даних про стан довкілля в регіоні.

*Функціональні вимоги:*

- збір даних за показниками екологічного стану регіону в реальному або наближеному до реального часі (температура, вологість, рівень забруднення повітря, якість води, радіаційний фон тощо);
- зберігання даних у надійній базі даних для подальшого аналізу (архівування, бекапи);
- аналітичну обробку даних, включно з:
  - статистичним аналізом,
  - інтервальним аналізом,
  - коротко- та середньостроковим прогнозуванням стану довкілля;
- візуалізацію даних: графіки, діаграми, інтерактивні карти;
- інтеграцію з зовнішніми джерелами (API державних та приватних екологічних станцій, сенсорні мережі, супутникові сервіси);
- зручний веб-доступ та мобільний доступ до інформації.

*Технічні вимоги:*

- клієнт-серверна архітектура: для забезпечення масштабованості та розділення функціональних рівнів;
- база даних: реляційна (PostgreSQL, MySQL) або гібридна (з елементами NoSQL, наприклад MongoDB);
- програмна реалізація:

- Backend: Python (Flask, Django), Node.js або Java;
- Frontend: React.js / Vue.js з використанням бібліотек візуалізації (D3.js, Chart.js, Leaflet, Mapbox);
- протоколи передачі даних: HTTPS, MQTT, REST API;
- вимоги безпеки: автентифікація, шифрування даних, контроль доступу;
- сумісність з платформами: браузери, Android / iOS;
- навантаження: підтримка мінімум 100 одночасних користувачів, можливість обробки ~10 тис. записів на годину.

*Бізнес-вимоги.* У якості прикладу таких вимог можна зазначити:

- Вимоги замовника (стейкхолдерів):
  - Надання об'єктивної інформації для екологічних служб, місцевих органів влади, громадськості;
  - Звітність та доказовість для екологічного контролю та політики;
  - Інформаційна підтримка при прийнятті рішень в галузях сільського господарства, логістики, урбаністики.
- Основне призначення:
  - прозорість – відкритий доступ до екологічної інформації;
  - оперативність – отримання критичних даних про загрози (забруднення, пожежі) у найкоротші терміни;
  - прогнозування ризиків – запобігання екологічним катастрофам;
  - зниження витрат – оптимізація витрат на екологічний аудит і контроль.
- Показники ефективності (KPI) системи:
  - час оновлення даних (latency) – не більше 2 хв;
  - точність прогнозу – не менше 85% за основними параметрами;
  - доступність системи – 99.5% uptime;
  - середній час реакції на аномальні значення – 1 хв.

## 2.2 Модель архітектури комп'ютерної системи екологічного моніторингу

Комп'ютерна система екологічного моніторингу має модульну структуру, що дозволяє розділити основні функціональні компоненти.

**Модуль збору даних** забезпечує прийом даних з різних джерел, якими можуть бути:

- апаратні датчики (температура, PM2.5, CO<sub>2</sub>, тощо);
- зовнішні API (наприклад, AirVisual, PurpleAir);
- генератор тестових даних (у режимі розробки).

Передбачена функціональність:

- приймання вхідних даних;
- форматування вхідних даних;
- перевірка валідності;
- передача в сховище або чергу обробки (через API або брокер повідомлень, наприклад, MQTT, Kafka).

**Сховище даних (БД).** Для реалізації бази даних пропонується використати SQLite, у майбутньому можлива заміна на PostgreSQL або MongoDB для масштабування.

Передбачена функціональність:

- структуроване зберігання даних;
- формування історії значень показників за датою/часом/геолокацією;
- передача даних на аналіз.

База даних є центральним модулем в архітектурі комп'ютерної системи моніторингу (рис. 2.1).



Рисунок 2.1 – Структура модулів комп’ютерної системи

Передбачається, що модулі комп’ютерної системи взаємодіють з базою даних у такій послідовності:

- вхідні дані записуються в БД;
- аналітичний модуль витягує дані для аналізу;
- інтерфейс користувача зчитує дані для візуалізації з бази даних і аналітичного модуля.

**Аналітичний модуль** призначений для виконання задач аналізу показників екологічного стану. Основні завдання:

- перевірка на перевищення ГДК – порівняння поточних значень із нормативними межами;
- статистичний аналіз – обчислення середніх, медіанних, максимальних/мінімальних значень;
- візуалізація динаміки – побудова графіків і теплових карт;
- виявлення аномалій і трендів: задачі класифікації екологічного стану, прогнозування показників забруднення, кластеризація даних.

**Веб-інтерфейс користувача** передбачає наявність таких інструментів або елементів:

- панель моніторингу;
- вибір регіону/часу;
- інтерактивні графіки, фільтри, завантаження звітів;
- авторизація, розмежування прав доступу.

### 2.3 Модель структури бази даних комп'ютерної системи

Для реалізації у роботі вибраний тип СУБД – SQLite через наступні причини:

- простота та легкість у розгортанні;
- підходить для вбудованих застосунків та прототипів;
- не потребує окремого серверу (self-contained);
- добре інтегрується з мовами Python, JavaScript тощо;
- може бути замінена на PostgreSQL або MongoDB для розподіленої або масштабованої системи.

База даних має одну основну таблицю `measurements` (табл. 2.1), кожен запис у якій містить інформацію про один замір показників екологічного стану середовища, і одну додаткову таблицю `devices` (табл. 2.2), яка призначена для ідентифікації джерела даних, що дозволяє фільтрувати дані по пристрою.

*Таблиця 2.1*

#### Структура таблиці `measurements` бази даних

Поле	Тип	Опис
<code>id</code>	INTEGER	Первинний ключ
<code>timestamp</code>	TEXT / DATETIME	Час запису
<code>temperature</code>	REAL	Температура (°C)
<code>humidity</code>	REAL	Вологість (%)
<code>co2</code>	REAL	CO <sub>2</sub> (ppm)
<code>pm25</code>	REAL	PM2.5 (мкг/м <sup>3</sup> )

Таблиця 2.2

## Структура таблиці devices

Поле	Тип	Опис
device_id	ТЕХТ	Унікальний ідентифікатор пристрою
location	ТЕХТ / DATETIME	Місце розташування (геолокація або опис)
type	ТЕХТ	Тип пристрою (датчик, API, симулятор)
status	ТЕХТ	Стан пристрою (active, inactive)

Таблиця measurements повинна мати зовнішній ключ device\_id, що зв'язує її з таблицею devices.

#### 2.4 Методи аналізу даних моніторингу екологічного стану

**Методи попередньої обробки даних (Preprocessing).** Попередня обробка є обов'язковим етапом для забезпечення коректності і повноти вхідних даних. У комп'ютерній системі моніторингу важливі такі етапи попередньої обробки як:

1) видалення або заповнення пропущених значень. Для цього використовуються такі підходи: заповнення середніми значеннями, інтерполяція або застосування моделей прогнозування для відновлення значень.

2) перетворення форматів необхідно для стандартизації форматів даних і одиниць вимірювань. У роботі використана стандартизація часу (UTC/локальний) та конвертація одиниць виміру до єдиного формату.

3) перевірка коректності значень, які були отримані на вході. У роботі передбачено виявлення аномальних значень, які свідчать про помилки введення (наприклад,  $PM_{2.5} > 1000$  мкг/м<sup>3</sup>). Можливо додавання фільтрації шуму або викидів з використанням ковзного середнього або фільтрів Калмана.

**Методи статистичного аналізу** використовуються для обчислення описової статистики, наприклад, середнього, медіани, мінімуму, максимуму, коефіцієнтів варіації та розмахів значень. Важливим є застосування

дисперсійного аналізу і визначення стандартних відхилень, оскільки дозволяється виявити стабільність середовища та періоди пікових навантажень (наприклад, вранці або ввечері).

Результати статистичного аналізу є основою для візуалізації даних, наприклад, для побудови гістограм, графіків трендів, теплових карт, динамічного перегляду значень показників в діапазоні дат.

Для аналізу і виявлення перевищення норм ГДК використовується порівняння значень показників концентрації речовин у середовищі з нормативами, наприклад: для CO<sub>2</sub> допустима концентрація до 1000 ppm, для PM<sub>2.5</sub> допустима концентрація до 35 мкг/м<sup>3</sup>. У разі перевищення формуються тригери для інформування або аварійного реагування.

**Методи ШІ та машинного навчання** можуть бути впроваджені в системи екологічного моніторингу для вирішення досить великого переліку задач.

- виявлення аномалій (Anomaly Detection) – несподіваних стрибків або падінь значень показників, несправностей у датчиках. Можливе використання методів: One-Class SVM, Isolation Forest, Autoencoders (глибокі нейромережі);

- прогнозування показників (Time Series Forecasting) – прогнозування рівня забруднення на кілька годин або днів наперед, планування дій на основі очікуваного погіршення стану. Моделі і методи ARIMA, SARIMA, LSTM (довготривала короткочасна пам'ять), Prophet (від Meta);

- кластеризація та групування даних – виявлення схожих ділянок за екологічними параметрами, групування даних для регіонального аналізу. Методи: K-means, DBSCAN;

- пояснювальна аналітика (Explainable AI) – розуміння, які саме параметри впливають на зміну стану середовища. Методи: SHAP, LIME.

## **Висновки за розділом 2**

Розроблена модель комп'ютерної системи екологічного моніторингу відповідає сучасним вимогам щодо збору, зберігання, аналізу та візуалізації даних. Її модульна архітектура забезпечує гнучкість і масштабованість, а використання технологій машинного навчання відкриває перспективи для прогнозування та виявлення аномалій. Система орієнтована на реальні потреби користувачів – від екологічних служб до громадськості – та здатна забезпечити оперативність, точність і прозорість екологічної інформації.

## РОЗДІЛ 3

### ПРОГРАМНА РЕАЛІЗАЦІЯ КОМП'ЮТЕРНОЇ СИСТЕМИ ТА ЇЇ ТЕСТУВАННЯ

#### 3.1 Обґрунтування вибору програмних технологій

Сучасні технології пропонують широкий вибір програмних мов, засобів, інструментів для програмної розробки. Для розробки комп'ютерної системи екологічного моніторингу необхідно обрати інструменти, які забезпечать зручну роботу з даними, гнучкість інтеграції, просту візуалізацію та ефективну обробку запитів.

Огляд поширених інструментів для подібних проєктів:

- Мови програмування:
  - Python – найбільш поширена для аналітики, ШІ, роботи з даними (pandas, NumPy, scikit-learn);
  - JavaScript (Node.js) – для серверної логіки та API;
  - Java/Kotlin – у великих проєктах, зокрема для Android-додатків;
  - Go, Rust – для продуктивних обчислень або мікросервісів.
- Бази даних:
  - SQLite – для локальних і невеликих проєктів;
  - PostgreSQL, MySQL – для масштабованих реляційних рішень;
  - MongoDB, InfluxDB – у випадках з неструктурованими або часовими даними;
  - Firebase – для мобільних рішень із синхронізацією.
- Інструменти аналітики та ML:
  - pandas, NumPy – обробка табличних даних;
  - scikit-learn, XGBoost – алгоритми машинного навчання;
  - TensorFlow, PyTorch – для нейронних мереж (при необхідності складних моделей).

- Візуалізація:
  - Matplotlib, Seaborn, Plotly – графіки й аналітичні візуалізації;
  - Leaflet.js, Mapbox, D3.js – карти та інтерактивні елементи;
  - Grafana – при інтеграції з базами моніторингу.
- Фреймворки для веб-розробки:
  - Flask, Django – Python-фреймворки для API та інтерфейсів;
  - React.js, Vue.js – фронтенд фреймворки для створення інтерактивних UI;
  - FastAPI – швидкий фреймворк для REST API на Python.
- Комунікаційні протоколи:
  - REST API, WebSocket, MQTT – для інтеграції між модулями, збору даних з сенсорів;
  - HTTPS, TLS – для безпечної передачі даних.

У таблиці 3.1 перелічені технології, вибрані для створення прототипу комп'ютерної системи екологічного моніторингу.

*Таблиця 3.1*

#### **Вибрані технології розробки прототипу комп'ютерної системи**

<b>Компонент</b>	<b>Технологія</b>	<b>Критерій вибору</b>
Мова розробки	Python	Простота роботи з даними, бібліотеки
Сховище	SQLite	Легка вбудована БД для невеликих систем
Аналітика	pandas, NumPy	Зручна обробка табличних даних
Візуалізація	Matplotlib	Побудова графіків
Інтерфейс	Flask + HTML	Легка веб-реалізація

Для реалізації модулю обробки даних використовується мова Python з бібліотеками: pandas – для обробки табличних даних; matplotlib, seaborn – для побудови графіків; sqlite3 – доступ до бази даних; datetime – для обробки часових міток.

Для реалізації користувацького інтерфейсу використаний Flask, бо дозволяє легко відтворити перегляд поточних значень показників, статистику, візуалізувати графіки.

## **3.2 Збір і збереження даних моніторингу**

### **3.2.1 Джерела і збір даних моніторингу**

Джерело даних для системи моніторингу – це один з ключових елементів. Можна реалізувати його двома основними способами.

1. Симульовані дані (імітація сенсорів) за допомогою програмних скриптів.

Переваги такого способу: повна контрольованість, незалежність від наявності інтернет зв'язку, легке масштабування та тестування.

Реалізація була здійснена за допомогою програмного скрипта `sensor_simulator.py` (додаток Г), який генерує випадкові значення показників повітря з реалістичним діапазоном:

- температура: 10–35 °C;
- вологість: 30–90 %;
- CO<sub>2</sub>: 300–1000 ppm;
- PM<sub>2.5</sub>: 0–100 мкг/м<sup>3</sup>.

Значення показників можна зберігати кожні 10 секунд або кожну хвилину, таким чином імітуючи реальні сенсори. Кожен запис, який зберігається до бази даних містить дату і час замірів (імітацій), значення показників температури, вологості, CO<sub>2</sub>, PM<sub>2.5</sub>, або за необхідності інших речовин.

2. Реальні джерела даних (API).

Переваги API: реальні дані, релевантні для регіону.

Недоліки: потрібен доступ до інтернету і ключі API, можуть бути обмеження по кількості запитів.

Існує кілька варіантів API для екологічного моніторингу, які наведені у табл. 3.2.

Таблиця 3.2

### API для екологічного моніторингу

Джерело	Тип даних	Посилання
OpenWeatherMap	Температура, вологість, PM2.5	<a href="https://openweathermap.org/api">https://openweathermap.org/api</a>
Luftdaten / Sensor.Community	PM10, PM2.5, температура	<a href="https://sensor.community/en/export/">https://sensor.community/en/export/</a>
IQAir (AirVisual)	Якість повітря (AQI, PM2.5)	<a href="https://www.iqair.com/world-air-quality">https://www.iqair.com/world-air-quality</a>
Air Quality Open Data Platform (ECMWF)	Модельні дані для Європи	<a href="https://atmosphere.copernicus.eu">https://atmosphere.copernicus.eu</a>

Збір значень показників екологічного стану із реального API (наприклад, OpenWeather або Sensor.Community) реалізується як періодичний запит (`requests.get`) до API. Отримані значення показників зберігаються до бази даних записами такої ж структури, як і для симульованих даних.

#### 3.2.2 Програмна реалізація бази даних

Для реалізації бази даних використовувалась проста база даних SQLite, оскільки вона вбудована і не потребує окремого серверу, також добре підходить для локальних додатків та демонстрації. База даних створюється та керується за допомогою стандартної бібліотеки `sqlite3`, яка інтегрована у Python.

Створення бази даних відбувалося у кілька етапів.

1. Підключення до бази даних:

```
import sqlite3
```

```
conn = sqlite3.connect("monitoring.db")
cursor = conn.cursor()
```

**2. Створення таблиці measurements для зберігання даних моніторингу – значень показників замірів екологічного стану:**

```
CREATE TABLE IF NOT EXISTS measurements (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    timestamp TEXT,
    temperature REAL,
    humidity REAL,
    co2_level REAL
);
```

**3. Додавання нового запису (наповнення даними):**

```
cursor.execute("""
    INSERT INTO measurements (timestamp, temperature, humidity,
co2_level)
    VALUES (?, ?, ?, ?)
""", (current_time, temp, hum, co2))
conn.commit()
```

**4. Отримання останніх значень:**

```
cursor.execute("SELECT * FROM measurements ORDER BY timestamp DESC
LIMIT 10")
latest_data = cursor.fetchall()
```

**5. Агрегація або побудова статистики за період:**

```
cursor.execute("""
    SELECT AVG(temperature), AVG(humidity), AVG(co2_level)
    FROM measurements
    WHERE timestamp BETWEEN ? AND ?
""", (start_time, end_time))
```

```
stats = cursor.fetchone()
```

Для зручності використання в Python, результати запитів обробляються з допомогою бібліотек pandas (наприклад, для перетворення у DataFrame) та datetime для роботи з часовими інтервалами.

### 3.3 Аналіз даних моніторингу екологічного стану регіону

У моделі комп'ютерної системи передбачений модуль аналітики, який забезпечує виконання задач статистичного аналізу, візуалізації, класифікації стану середовища. Розглянемо приклади роботи даного модуля.

1. Формування статистики за певний період часу (Summary Statistics) дозволяє отримати розуміння загальної картини стану середовища.

Наприклад, в результаті запиту за 10 днів були отримані значення, наведені у табл. 3.3.

Таблиця 3.3

#### Результат статистичного аналізу

Показник	Температура (°C)	Вологість (%)	CO <sub>2</sub> (ppm)
Середнє	22.4	55.1	824
Мінімум	17.0	42.3	410
Максимум	29.7	71.5	1200
Ст. відхилення	3.1	7.2	186

За результатами статистичного аналізу видно, що температурні показники та рівень вологості стабільні. Однак рівень CO<sub>2</sub> підходить до гранично допустимого значення (1000 ppm), що потребує уваги.

2. Побудова графіків тренду (Trend Analysis) необхідно для розуміння змін показників у часі. На рис. 3.1 показаний графік зміну рівня CO<sub>2</sub> у часі.

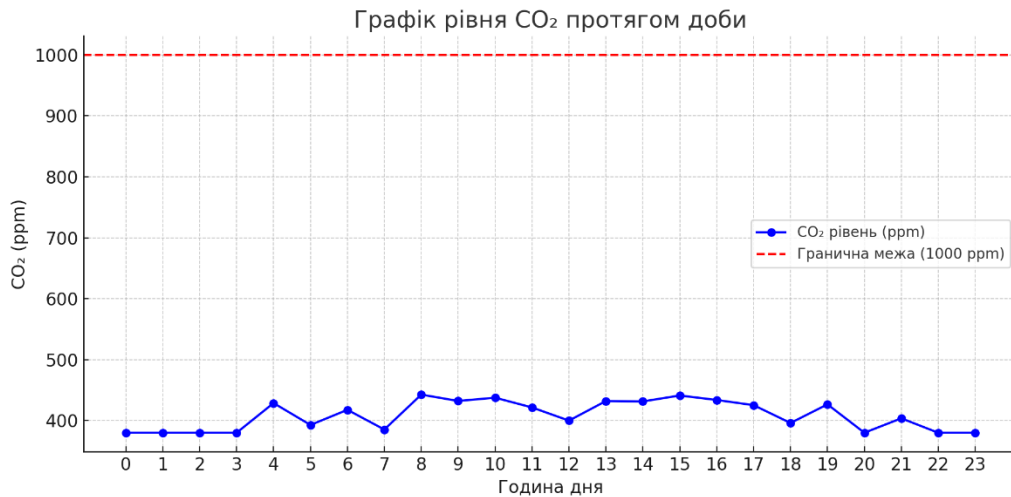


Рисунок 3.1 – Графік рівня CO<sub>2</sub>

Графік демонструє поступове зростання рівня CO<sub>2</sub> у денний час, можливо через людську активність або закриті приміщення. Червона лінія — допустима межа (1000 ppm).

3. Виявлення перевищень ГДК передбачає обчислення кількості перевищень і визначення моментів перевищення допустимих концентрацій.

Запит може мати такий код:

```
exceeds = df[df['co2_level'] > 1000]
print(f"Кількість перевищень ГДК: {len(exceeds)}")
print(exceeds[['timestamp', 'co2_level']])
```

У періоді моніторингу зафіксовано, наприклад, 7 перевищень ГДК CO<sub>2</sub>. Передбачається можливість налаштувань для інших показників, наприклад, перевірки рівня вентиляції або забруднення шкідливими речовинами.

4. Побудова теплової карти (Heatmap) коливань температури допомагає візуалізувати щоденні коливання температури (рис. 3.2).

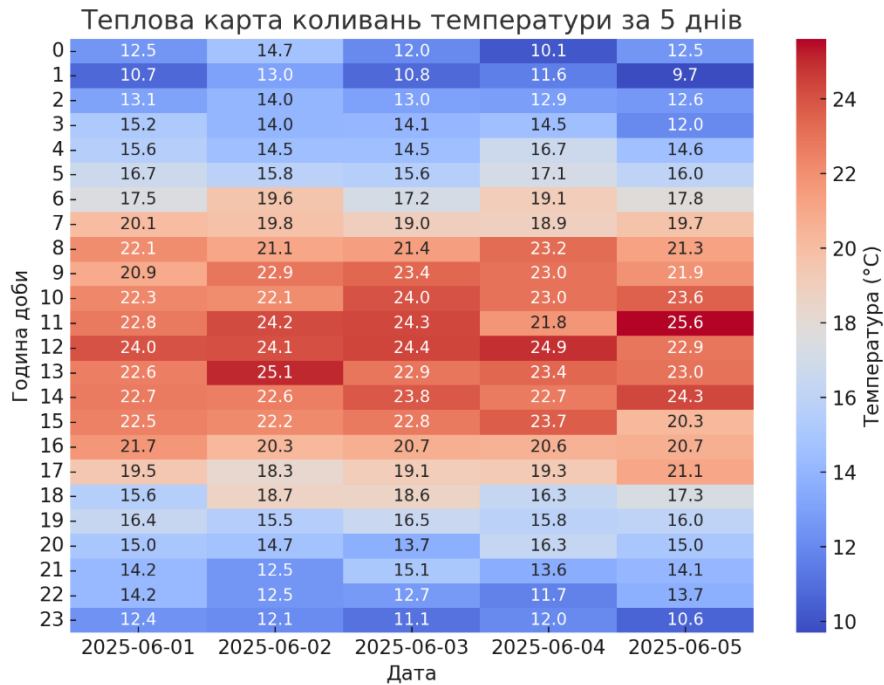


Рисунок 3.2 – Теплова карта коливань температури

Картина на рис. 3.2 показує типовий добовий цикл температури — мінімум вночі, максимум у другій половині дня. Може використовуватись для виявлення незвичних теплових аномалій.

5. Визначення індексу якості повітря передбачає розрахунок умовного індексу AQI, який використовується для класифікації стану повітря на основі рівня CO<sub>2</sub> та PM<sub>2.5</sub> (за наявності).

Алгоритм пошуку індексу AQI представлений у вигляді коду відповідного методу:

```
def classify_air_quality(co2):
    if co2 <= 600:
        return "Добре"
    elif co2 <= 1000:
        return "Задовільно"
    elif co2 <= 1500:
        return "Погіршене"
```

```
else:
    return "Небезпечне"
```

Нехай 65% вимірювань – у категорії "Добре", 30% – "Задовільно", 5% – "Погіршене". У такому разі необхідно слідкувати за динамікою "погіршених" значень.

### 3.4 Тестування комп'ютерної системи моніторингу екологічного стану

Коротке керівництво користувачеві.

#### 1. Запуск системи

- Встановити Python та необхідні бібліотеки (Flask, sqlite3, pandas, matplotlib).

- Запустити файл app.py командою:

```
python app.py
```

- Відкрити браузер і перейти за адресою: <http://127.0.0.1:5000/>

2. Головна сторінка відображає останні екологічні заміри у вигляді таблиці (рис. 3.3). Таблиця містить:

- дату і час заміру;
- температуру, вологість, CO<sub>2</sub>, PM<sub>2.5</sub>;
- значення, що перевищують ГДК, виділені кольором.

Дата й час	Температура	Вологість	CO <sub>2</sub> (ppm)	PM <sub>2,5</sub> (мкг/м <sup>3</sup> )
25.04 9:00	19,0°C	42 %	650	10
25.04 10:00	20,2°C	40 %	600	15
25.04 10:45	20,0°C	41 %	700	16
25.04 11:30	21,0°C	40 %	700	16
25.04 12:00	22,1°C	85 %	850	33

Рисунок 3.3 – Таблиця показників екологічного стану

Користувачу надається можливість візуалізувати значення показників за певний період моніторингу. Наприклад, користувач вибирає для візуалізації два показники CO<sub>2</sub>, PM<sub>2.5</sub>, задає період збору даних. Система надає графік зміни цих показників (рис. 3.4) і у разі перевищення допустимої границі виводить повідомлення про це.



Рисунок 3.4 – Візуалізація показників моніторингу

Також користувач може подивитися індекс якості повітря (рис. 3.5), який може мати один з чотирьох значень. У разі перевищення концентрації рівня CO<sub>2</sub> система покаже AQI – «Погіршене» або «Небезпечне».

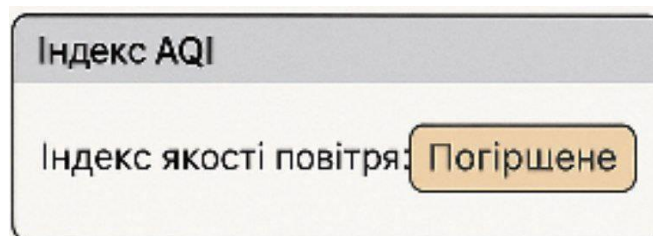


Рисунок 3.5 – Значення індексу якості повітря

### **Висновки за розділом 3**

У результаті розробки прототипу комп'ютерної системи моніторингу екологічного стану було обґрунтовано вибір сучасних технологій, реалізовано збирання, зберігання та базову аналітику екологічних даних. Для програмної реалізації обрано мову Python та фреймворк Flask, що забезпечили швидку побудову веб-інтерфейсу, інтеграцію з базою даних SQLite та можливість масштабування системи в майбутньому.

Проведено імітаційне тестування з використанням симульованих даних, продемонстровано ефективність аналізу та візуалізації. Інтерфейс користувача забезпечує зручний доступ до даних у режимі наближеному до реального часу.

Розроблена система є гнучкою, розширюваною та придатною для подальшого впровадження в локальних або регіональних екологічних проєктах.

## ВИСНОВКИ

У кваліфікаційній роботі було розроблено модель комп'ютерної системи для моніторингу екологічного стану регіону, яка дозволяє ефективно збирати, обробляти, зберігати та аналізувати дані про стан навколишнього середовища. Проведено повний цикл проектування – від аналізу предметної області до реалізації функціонального прототипу з веб-інтерфейсом.

Робота включала:

- аналіз існуючих систем (AirVisual, PurpleAir, EcoMonitoring, AWS) та визначення їх переваг і обмежень;
- формування функціональних, технічних та бізнес-вимог до майбутньої системи;
- проектування архітектури у вигляді модулів (збір, сховище, аналітика, інтерфейс);
- побудову структури бази даних з урахуванням можливості масштабування;
- реалізацію основних функцій у середовищі Python із використанням Flask, SQLite, pandas, matplotlib;
- тестування роботи системи на основі згенерованих даних;
- візуалізацію результатів та інтерпретацію екологічних показників;

У межах проєкту реалізовано коротке керівництво користувача та підготовлено графічні інтерфейси для перегляду даних.

Особливу увагу приділено аналітичним методам обробки інформації, зокрема – статистичному аналізу, перевірці на перевищення ГДК та перспективам впровадження алгоритмів машинного навчання (прогнозування, виявлення аномалій).

У результаті створено працездатний програмний прототип, який може бути використаний як основа для повнофункціональної системи екологічного

моніторингу на рівні громади, навчального закладу чи окремого населеного пункту. Робота підтверджує можливість побудови ефективних локальних ІТ-рішень для підтримки екологічної безпеки на базі відкритих технологій і простих апаратних засобів.

## ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ ISO 37120:2017 Стійкий розвиток громад — Показники міських служб та якості життя.
2. ДСанПіН 3.3.2.007-98 Гігієнічні вимоги до ПЕОМ та організації праці.
3. ISO 14001:2015 Environmental management systems — Requirements with guidance for use.
4. Постанова КМУ №827 від 28.10.2021 «Про затвердження переліку показників якості довкілля».
5. Грабовий П.М. Основи екологічного моніторингу. — К.: Либідь, 2017.
6. Мельниченко О.М. Інформаційні системи моніторингу навколишнього середовища. — Вінниця: ВНТУ, 2020.
7. Власюк В.С. Технології аналізу екологічних даних. — Львів: ЛНУ, 2021.
8. Козак Ю.П. Основи баз даних: навч. посіб. — К.: КНЕУ, 2020.
9. Шеремет А.О. Використання Python для обробки екологічних даних // Електроніка та приладобудування. — 2021. — №2. — С. 45–49.
10. McKinney W. Python for Data Analysis. — O'Reilly Media, 2022.
11. Van Rossum G. The Python Tutorial. — Python Software Foundation, 2023. [Електронний ресурс]. — Режим доступу: <https://docs.python.org>
12. SQLite Documentation. — [Електронний ресурс]. — Режим доступу: <https://www.sqlite.org/docs.html>
13. Flask Documentation. — [Електронний ресурс]. — Режим доступу: <https://flask.palletsprojects.com/>
14. IQAir. AirVisual – Air Quality Monitoring System. — [Електронний ресурс]. — Режим доступу: <https://www.iqair.com>
15. PurpleAir. — [Електронний ресурс]. — Режим доступу: <https://map.purpleair.com>

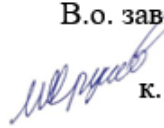
- 16.EcoMonitoring. — [Электронный ресурс]. — Режим доступа:  
<https://www.ecomonitoring.com>
- 17.Vaisala AWS310 Automatic Weather Station. — [Электронный ресурс]. —  
Режим доступа: <https://www.vaisala.com/en/aws310>
- 18.Scikit-learn User Guide. — [Электронный ресурс]. — Режим доступа:  
<https://scikit-learn.org>
- 19.Plotly Python Graphing Library. — [Электронный ресурс]. — Режим  
доступа: <https://plotly.com/python>
- 20.Pandas Documentation. — [Электронный ресурс]. — Режим доступа:  
<https://pandas.pydata.org/docs/>

## ДОДАТКИ

### Додаток А

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Харківський національний університет імені В. Н. Каразіна

Навчально-науковий інститут комп'ютерних наук та штучного інтелекту  
Кафедра комп'ютерних систем та робототехніки  
Рівень вищої освіти (освітньо-кваліфікаційний рівень) **Бакалавр**  
Галузь знань: 12 – Інформаційні технології  
Спеціальність: 123 «Комп'ютерна інженерія»  
Освітня програма «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ  
В.о. завідувача кафедри комп'ютерних  
систем та робототехніки  
 к. ф.-м. н., доц. ХРУСЛОВ М. М.  
«02» жовтня 2024 року

### ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

**АЗАРЕНКОВОЇ Олександр Андріївни**  
(прізвище, ім'я, по батькові студента)

1. Тема роботи **«Модель комп'ютерної системи моніторингу екологічного стану регіону»**

керівник роботи **Стрілець Вікторія Євгенівна, кандидат технічних наук**  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від **16 квітня 2025 року № 4101-5/962**

2. Строк подання студентом роботи **30 травня 2025 року**

3. Перелік питань, які потрібно розробити:

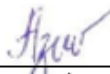
1. Аналіз існуючих технологій і підходів до вирішення задач екологічного моніторингу.
2. Створення моделі комп'ютерної системи моніторингу екологічного стану окремого регіону або місцевості.
3. Створення прототипу комп'ютерної системи моніторингу екологічного стану.
4. Тестування прототипу та аналіз подальших можливостей розвитку запропонованої моделі.

## 4. План роботи

№ з/п	Назви етапів роботи	Термін виконання етапів роботи
1	Пошук наукової літератури за темою роботи	02.10.2024 - 24.10.2024
2	Аналіз технологій і підходів до вирішення задач екологічного моніторингу.	02.10.2024 - 24.10.2024
3	Огляд методів аналізу даних, машинного навчання і ШІ для систем екологічного моніторингу.	25.10.2024 - 09.11.2024
4	Розробка моделі комп'ютерної системи моніторингу екологічного стану окремого регіону або місцевості.	10.11.2024 - 24.11.2024
5	Обґрунтування технічних рішень і програмних засобів для розробки прототипу комп'ютерної системи.	25.10.2024 - 08.11.2024
6	Практична реалізація прототипу комп'ютерної системи моніторингу екологічного стану	09.11.2024 - 29.11.2024
7	Тестування та оцінка ефективності моделі комп'ютерної системи моніторингу екологічного стану.	01.01.2025 - 01.02.2025
8	Аналіз можливостей подальшого вдосконалення запропонованої моделі.	01.03.2025 - 30.04.2025-
9	Підготовка і оформлення звітних матеріалів та додатків кваліфікаційної роботи. Оформлення списку літератури	01.04.2025 - 30.04.2025-
10	Оформлення пояснювальної записки кваліфікаційної роботи відповідно вимогам до звітів про НДР.	01.05.2025 - 30.05.2025-
11	Оформлення звіту про переддипломну практику	01.05.2025 - 30.05.2025-
12	Представлення кваліфікаційної роботи керівнику та рецензенту	30.05.2025

5. Дата видачі завдання *02 жовтня 2024 року.*

Студент

Азаренкова О.А.
  
 \_\_\_\_\_  
 підпис

Керівник роботи

Стрілець В.Є.
  
 \_\_\_\_\_  
 підпис

## Додаток Б

Затверджую

\_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 2025 р.

Технічне завдання  
на розробку програмного виробу  
«МОДЕЛЬ КОМП'ЮТЕРНОЇ СИСТЕМИ МОНІТОРИНГУ ЕКОЛОГІЧНОГО  
СТАНУ РЕГІОНУ»

Назва розділу	Назва і зміст підрозділу
1. Вступ	1.1. Назва програмного виробу – Модель комп'ютерної системи моніторингу екологічного стану регіону. 1.2. Галузь застосування – Інформаційні технології.
2. Підстава для розробки	2.1. Навчальний план за спеціальністю 123 – Комп'ютерна інженерія 2.2. Завдання на кваліфікаційну роботу бакалавра № <b>4101-5/962</b> від «16» квітня 2025 (представити як Додаток А до пояснювальної записки до кваліфікаційної роботи).
3. Призначення розробки	3.1. Мета розробки: підвищити якість прогнозування показників екологічного стану окремого регіону через застосування підходів штучного інтелекту та машинного навчання. 3.2. Призначення розробки: для створення комп'ютерної системи, яка автоматично збирає, аналізує та візуалізує екологічні дані регіону, виявляє перевищення ГДК, прогнозує зміни стану довкілля за допомогою ШІ та забезпечує зручний доступ до цієї інформації для влади, екологів і громадськості. 3.3. Початкові дані для розробки: параметри стану повітря (температура, вологість, CO <sub>2</sub> , PM <sub>2.5</sub> ), гранично допустимі концентрації, імітовані та API-дані, а також вимоги до зберігання й обробки в базі даних для подальшого аналізу й візуалізації.
4. Технічні вимоги до програмного виробу	1) Архітектура: клієнт-серверна, з можливістю масштабування. 2) База даних: SQLite (з можливістю переходу на PostgreSQL або MongoDB). 3) Мови програмування: Python для бекенду, HTML/Flask для інтерфейсу. 4) Протоколи передачі даних: HTTPS, REST API, MQTT.

	<p>5) Інтерфейс: веб-доступ з інтерактивною візуалізацією (графіки, таблиці).</p> <p>6) Безпека: автентифікація, шифрування, контроль доступу.</p> <p>7) Продуктивність: обробка до 10 000 записів/год, не менше 100 користувачів.</p>
<p>5. Вимоги до програмної документації.</p>	<p>Програмною документацією до виробу «Модель комп'ютерної системи моніторингу екологічного стану регіону» вважати:</p> <p>1) Справжнє Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до дипломної роботи).</p> <p>2) Програму і методику випробувань розробленого програмного виробу (представити у вигляді Додатку В до пояснювальної записки до дипломної роботи).</p> <p>3) Опис програмного виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи).</p>
<p>6. Техніко-економічні показники</p>	<p>В даному розділі можуть бути представлені:</p> <p>1) Справжнє Технічне завдання на розробку програмного виробу (представити у вигляді Додатку Б до пояснювальної записки до дипломної роботи).</p> <p>2) Методику розрахунку інформативності змінних стану (у вигляді глав 3.2 та 3.3 пояснювальної записки до кваліфікаційної роботи).</p> <p>3) Опис виробу (представити в розділі 3 пояснювальної записки до кваліфікаційної роботи)</p>
<p>7. Стадії і етапи розробки</p>	<p>1. Пошук наукової літератури за темою роботи</p> <p>2. Аналіз технологій і підходів до вирішення задач екологічного моніторингу.</p> <p>3. Огляд методів аналізу даних, машинного навчання і ШІ для систем екологічного моніторингу.</p> <p>4. Розробка моделі комп'ютерної системи моніторингу екологічного стану окремого регіону або місцевості.</p> <p>5. Обґрунтування технічних рішень і програмних засобів для розробки прототипу комп'ютерної системи.</p> <p>6. Практична реалізація прототипу комп'ютерної системи моніторингу екологічного стану</p> <p>7. Тестування та оцінка ефективності моделі комп'ютерної системи моніторингу екологічного стану.</p> <p>8. Аналіз можливостей подальшого вдосконалення запропонованої моделі.</p> <p>9. Підготовка і оформлення звітних матеріалів та додатків кваліфікаційної роботи. Оформлення списку літератури</p>

8. Порядок контролю і приймання	1) Перевірку ходу розробки програмного виробу керівнику робіт виконувати раз в 3 тижні. 2) Випробування програмного продукту провести відповідно до програми та методики випробувань на базі комп'ютерного класу. 3) Захист розробленої моделі провести на засіданні Атестаційної комісії. 4) Пояснювальну записку подати в електронному вигляді в 1 примірника.
---------------------------------	---

Виконавець

студентка групи К1-41

Азаренкова О.А.



Замовник

канд. техн. наук, доц.

Стрілець В.Є.



**Додаток В****Програма і методика випробувань програмного виробу**

«Модель комп'ютерної системи моніторингу екологічного стану регіону»

**1. Об'єкт випробувань**

1. Назва програмного виробу : «Модель комп'ютерної системи моніторингу екологічного стану регіону»

2. Галузь застосування : Інформаційні технології.

3. Перераховані відомості запозичуються з відповідних розділів Технічного завдання.

**2. Мета випробувань**

Перевірка відповідності функціональності програмної реалізації системи заявленим функціональним можливостям в технічному завданні (Додаток Б до пояснювальної записки до кваліфікаційної роботи).

**3. Загальні положення****1. Підстави для проведення випробувань**

Підставою для проведення випробувань є наказ про призначення атестаційної комісії.

**2. Місце і тривалість випробувань**

Приймальні (приймально-здавальні) випробування проводяться на базі комп'ютерного класу кафедри в період роботи атестаційної комісії.

**3. Обсяг випробувань**

Приймальні випробування програмного виробу проводяться в обсязі відповідному цієї програми і методики випробувань.

**4. Організації, які беруть участь у випробуваннях**

Приймальні випробування проводяться атестаційною комісією напередодні засідання (або в процесі засідання) за участю Замовника, Виконавця та інших осіб, присутніх на засіданні.

#### **4. Вимоги до програми або програмного виробу**

##### 1. Функціональні вимоги:

- Збір даних з сенсорів або API (CO<sub>2</sub>, PM2.5, температура, вологість).
- Зберігання даних у базі (SQLite). Виявлення перевищень ГДК.
- Розрахунок індексу якості повітря (AQI).
- Візуалізація даних у вигляді графіків і таблиць.
- Формування звітів і інтерфейс користувача.

##### 2. Нефункціональні вимоги:

- Простота встановлення (локальний запуск без сервера).
- Інтуїтивний веб-інтерфейс.
- Швидкий доступ до даних (затримка до 2 хв).
- Доступність: не менше 99.5% часу.
- Безпека: контроль доступу, захист даних.

##### 3. Технічні вимоги:

- Python, Flask, SQLite, pandas, matplotlib.
- Підтримка мінімум 100 одночасних користувачів.
- Можливість масштабування (перехід на PostgreSQL).

#### **5. Вимоги до програмної документації**

Програмою документацією до виробу «Модель комп'ютерної системи моніторингу екологічного стану регіону» вважати:

1. Програмою документацією щодо розроблюваного програмного продукту вважати:

2. справжнє технічне завдання на розробку програми (представити як Додаток Б до пояснювальної записки до кваліфікаційної роботи);

3. Програму і методику випробувань розробленої програми (представити як Додаток В до пояснювальної записки до кваліфікаційної роботи);

4. Рекомендацій щодо застосування створеної програмної стандартизації у проектах (представити в Розділі 3 пояснювальної записки до кваліфікаційної роботи).

## **6. Засоби і порядок випробувань**

### **6.1 Засоби випробувань**

Для проведення випробувань необхідний персональний комп'ютер.

### **6.2 Порядок проведення випробувань**

Як правило, випробування проводяться в два етапи:

-ознайомчий (1-й етап);

-випробування програмного виробу (2-й етап).

Перелік перевірок, що проводяться на 1 етапі випробувань, включає в себе:

1. Перевірку комплектності програмної документації.
2. Перевірка комплектності складу програмної документації здійснюється за критерієм наявності зазначеної в ТЗ документації.
3. Перевірку комплектності складу технічних і програмних засобів.
4. Методику проведення перевірок на 1 етапі випробувань.
5. Якість програмної документації перевіряється на відповідність вимогам стандартів ДСТУ.

Перелік перевірок, що проводяться на 2 етапі випробувань, включає в себе:

1. Перевірку відповідності технічних характеристик програми вимогам технічного завдання;

2. Перевірку ступеня виконання функціональних вимог до програми;
3. Методику проведення перевірок, що входять до переліку по 2 етапу випробувань.

Тест 1. Виведення значень показників на головну сторінку.

Дата й час	Температура	Вологість	CO <sub>2</sub> (ppm)	PM <sub>2,5</sub> (мкг/м <sup>3</sup> )
25.04 9:00	19,0°C	42 %	650	10
25.04 10:00	20,2°C	40 %	600	15
25.04 10:45	20,0°C	41 %	700	16
25.04 11:30	21,0°C	40 %	700	16
25.04 12:00	22,1°C	85 %	850	33

Рисунок В.1 – Результат моніторингу за показниками

Тест 2. Візуалізація динаміки зміни показників

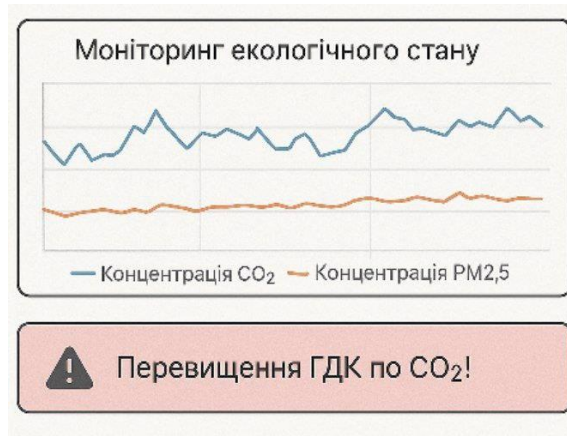


Рисунок В.2 – Графік зміни показників у часі

Тест 3. Визначення індексу якості повітря.

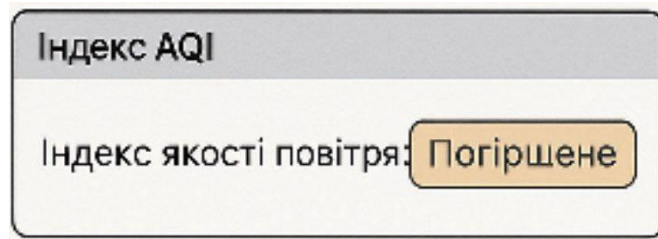


Рисунок В.3 – Значення індексу якості повітря

**Висновки:** Випробування пройшло успішно.

Виконавець: студентка групи КІ-41, Азаренкова О.А.

*Азаренкова*

## Фрагменти лістингу коду

### Користувацький інтерфейс

```
from flask import Flask, render_template
import pandas as pd
import sqlite3

app = Flask(__name__)

@app.route('/')
def index():
    conn = sqlite3.connect('data/eco_data.db')
    df = pd.read_sql("SELECT * FROM sensors ORDER BY timestamp DESC LIMIT 20",
conn)
    conn.close()
    return render_template("index.html", data=df.to_dict(orient='records'))
```

### Створення таблиці бази даних і збереження даних

```
CREATE TABLE sensors (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    timestamp TEXT,
    temperature REAL,
    humidity REAL,
    co2 REAL,
    pm25 REAL
);

import random, sqlite3
from datetime import datetime

conn = sqlite3.connect('data/eco_data.db')
cursor = conn.cursor()
```

```

for _ in range(10):
    cursor.execute("""
        INSERT INTO sensors (timestamp, temperature, humidity, co2, pm25)
        VALUES (?, ?, ?, ?, ?)
    """, (
        datetime.now().isoformat(),
        round(random.uniform(18, 30), 1),
        round(random.uniform(40, 70), 1),
        random.randint(400, 1300),
        round(random.uniform(5, 80), 1)
    ))

conn.commit()
conn.close()

```

## Генерація даних (імітація сенсорів)

```

import random

def generate_fake_data():
    return {
        "temperature": round(random.uniform(15, 35), 1),
        "humidity": round(random.uniform(40, 80), 1),
        "co2": round(random.uniform(350, 1000), 1),
        "pm25": round(random.uniform(5, 60), 1)
    }

```

## Збір даних і запис у базу даних

```

def save_to_db(temp, hum, co2, pm25):
    conn = sqlite3.connect("data/eco_data.db")
    cursor = conn.cursor()

    cursor.execute(

```

```

        "INSERT INTO sensors (timestamp, temperature, humidity, co2, pm25)
VALUES (?, ?, ?, ?, ?)",
        (datetime.now(), temp, hum, co2, pm25)
    )

```

## Автоматичний збір кожні N секунд

```

import time
import random

while True:
    # Симуляція
    temp = random.uniform(20, 30)
    hum = random.uniform(40, 70)
    co2 = random.uniform(350, 1000)
    pm25 = random.uniform(5, 40)

    save_to_db(temp, hum, co2, pm25)
    time.sleep(10) # кожні 10 сек

```

## Перевірка і перегляд збережених даних

```

def fetch_data(limit=10):
    conn = sqlite3.connect("data/eco_data.db")
    cursor = conn.cursor()
    cursor.execute("SELECT * FROM sensors ORDER BY timestamp DESC LIMIT ?",
(limit,))
    data = cursor.fetchall()
    conn.close()
    return data

```

## Запит на статистичний аналіз

```

import pandas as pd
import sqlite3

```

```

conn = sqlite3.connect("monitoring.db")
df = pd.read_sql_query("""
    SELECT timestamp, temperature, humidity, co2_level
    FROM measurements
    WHERE timestamp BETWEEN '2025-05-01' AND '2025-05-10'
""", conn)

summary = df.describe()
print(summary)

```

## Побудова графіку тренду

```

import matplotlib.pyplot as plt

df['timestamp'] = pd.to_datetime(df['timestamp'])
df.set_index('timestamp')['co2_level'].plot(figsize=(10, 5), title='Рівень CO2 з часом')

plt.axhline(y=1000, color='red', linestyle='--', label='ГДК')

plt.ylabel('ppm')
plt.legend()
plt.show()

```

## Побудова теплової карти

```

import seaborn as sns

df['date'] = df['timestamp'].dt.date
df['hour'] = df['timestamp'].dt.hour

pivot_table = df.pivot_table(values='temperature', index='hour', columns='date')
sns.heatmap(pivot_table, cmap='coolwarm', annot=False)

plt.title("Теплова карта температур за днями")
plt.ylabel("Година дня")
plt.xlabel("Дата")
plt.show()

```